IN THE SPECIFICATION

On page 1, line 9, replace the paragraph with:

This application relates to the inventors' concurrently filed and commonly assigned U. S. Patent Application 10/077,345, attorney docket number POU920020010US1, U. S. Patent Application 10/077,246, attorney docket number POU920020012US1, U. S. Patent Application 10/077,320, attorney docket number POU920020011US1, U. S. Patent Application 10/077,201, attorney docket number POU920020013US1, and U. S. Patent Application 10/077,129, attorney docket number POU920020014US1. All of these applications were filed on February 15, 2002 and are commonly assigned to International Business Machines Corporation.

On page 3, line 22, replace the paragraph with:

An exemplary file system data structure 400 is illustrated in FIG. 4 that contains two inodes, ~~one~~ an inode 402 in the active file system and ~~one~~ a snapshot inode 408 in a snapshot dataset, that each have disk addresses 404, 410 that point to the same data block 406. The existence of multiple references to a single data block within the original file system impacts the requirements of the original file system. File systems that utilize snapshots that each store a reference to an original data block must maintain an indication of each reference to that data block in order to determine if the data block is in-use or free. Without multiple references, a single bit is able to indicate if a data block is in-use or free. With the multiple references, multiple bits are required to track the multiple references and ensure that no references exist to the data block prior to declaring the data block "free." This need to track the multiple references complicates the operation of the file system, limits the total number of snapshots, and also complicates, or renders impossible, the implementation of such snapshot system with file systems that do not support tracking multiple references to a data block.

On page 15, line 4, replace the paragraph with:

FIG. 2B illustrates a second exemplary data structure set 250 that describes data files within the file system that is used in conjunction with the exemplary embodiments of the present invention. The exemplary set 200 of data structures is used for small files where the disk address data 206 is able to be stored within the inode itself. The second exemplary set 250 of data structures is used for larger files, where the disk address data 256 of inode 252 stores pointers to one or more indirect blocks 258 that each contain one or more pointers, such as the indirect block first disk address 260 and indirect block second disk address 262, that point to either the data blocks, such as first data block 264 and second data block 266, or to other indirect blocks that contain further disk addressing information. The inode 252 further includes metadata 254.

On page 15, line 26, replace the paragraph with:

The data for the first directory record 304 contains a file name 306 and inode number 308. The directories 302 are managed by the file system and the record 304 maps the user assigned file name 306 to the inode number 308 that uniquely identifies the inode 316 for the file associated with this directory entry 304. The data for the second directory record 310 similarly contains a file name 312 and inode number 314 that uniquely identifies inode 318. These directory entries are sometimes referred to as links in the exemplary embodiments. The links in these embodiments point to user files, other directories, or other file system objects not described herein. The directories of the exemplary embodiments impose a hierarchical naming structure over the files in the file system. The root of the hierarchy is the root directory of the file system. Files and directories within the file system are accessible from the root directory or a subdirectory from the root. A file is able to have more than one directory entry, and thus more than one name, for the user to access it.

On page 35, line 20, replace the paragraph with:

Specialized processing is performed by the exemplary embodiments of the present invention to modify, or update, a snapshot when there are more than

one snapshot present. In an example of a file system that has a series of snapshots ~~600~~ 620, as illustrated in FIG. 6B, identified as S1, ..., Si-1, Si, Si+1, ..., Sn, wherein the snapshots are ordered by decreasing age, e.g., S1 is the oldest snapshot, and Sn is the most recent snapshot. The snapshot update processing as performed by an exemplary embodiment of the present invention is illustrated in control flow 720 of FIG. 7B.

On page 37, line 14, replace the paragraph with:

Once the data to be modified is copied into the snapshot dataset associated with snapshot Si and an appropriate capture of the data is included in snapshot Si-1, the data within snapshot Si is modified, at step 732, as determined by the user or other processing within the exemplary embodiment. The processing then stops at step 734.